End-to-End ML Project
Model Tuning

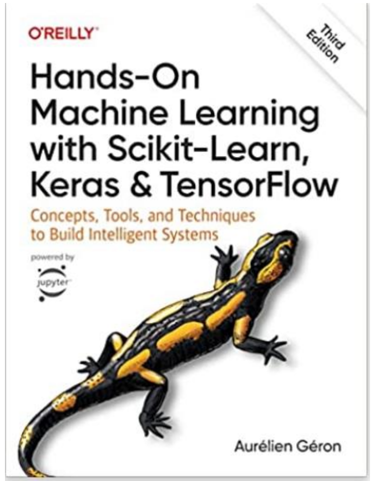
BA810: Supervised Machine Learning

Nachiketa Sahoo

Recap

End-to-End ML Project

1. Load and explore data, training-test split
2. Clean and consider transformations
3. Create pipelines
4. Evaluate various predictive models
5. Finetune hyperparameters of the most promising model
6. Estimate error on unused test-data


O'REILLY
Third Edition
Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
Concepts, Tools, and Techniques to Build Intelligent Systems
powered by jupyter
Aurélien Géron

Chapter 2

Advanced Model Tuning

Learn where to search

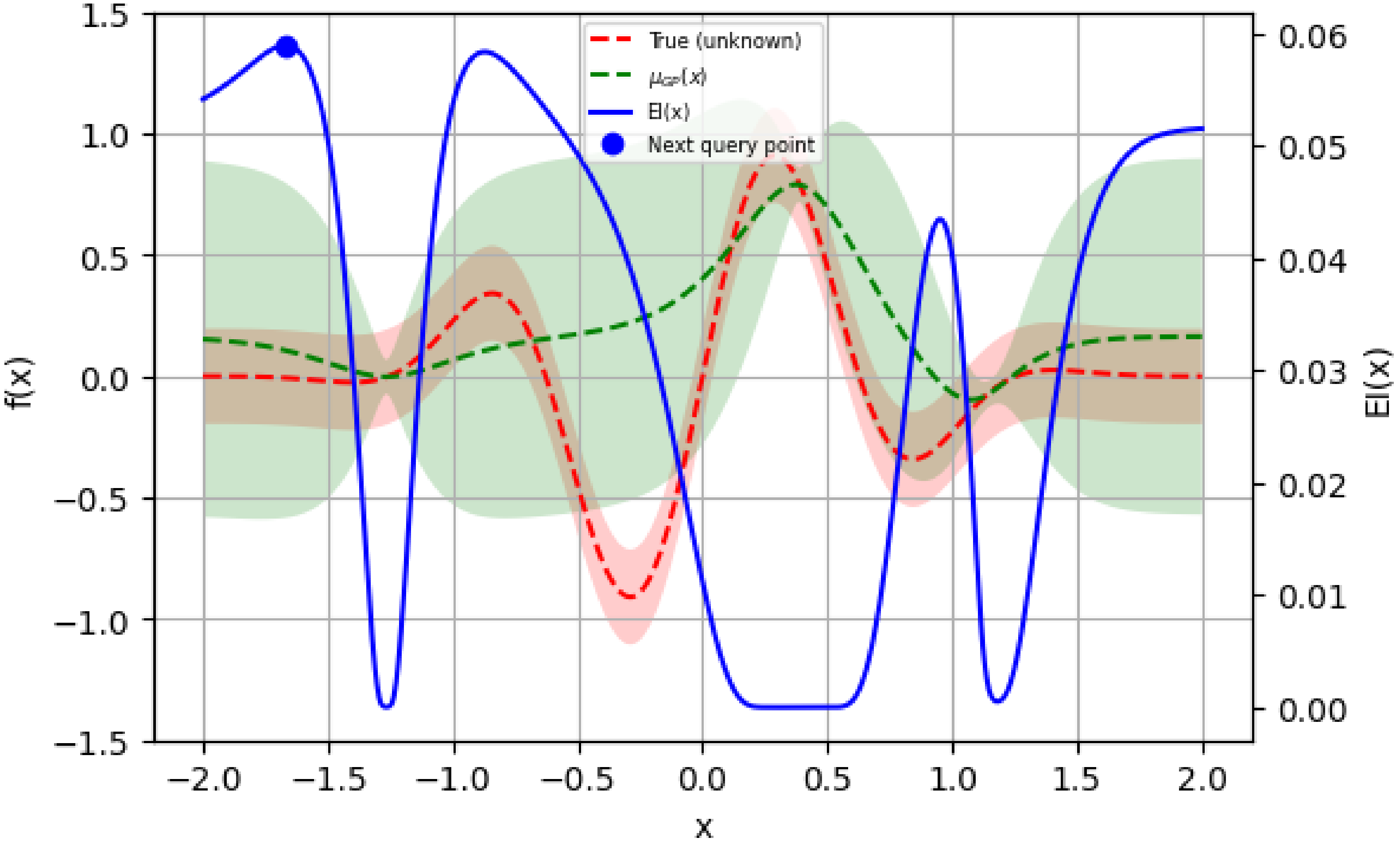
Recap

Model tuning (choosing hyper-parameters)

- **Grid search:** pre-commit to all possible combinations of specific hyperparameter values, evaluate model at each combination
  - Usable with a few hyper parameters, taking values that can be exhaustively explored
- **Random search:** for each trial, randomly draw values of each parameter; then evaluate the model at the trial value
  - Large number of parameters with large ranges that cannot be exhaustively evaluated
- **Halving searches:** deploy compute smartly
  - Grid/random strategy generate candidates
  - Evaluate all cheaply (with less data), more promising ones thoroughly (more data)
  - Speeds up search, risks missing good hyper parameter values due to premature elimination
- Can we *generate* promising candidates?
  - Based on a few candidates learn which regions have better accuracy and sample there
  - Or sample in regions with great uncertainty
    - There might be a hidden gem there
  - Exploit-explore …

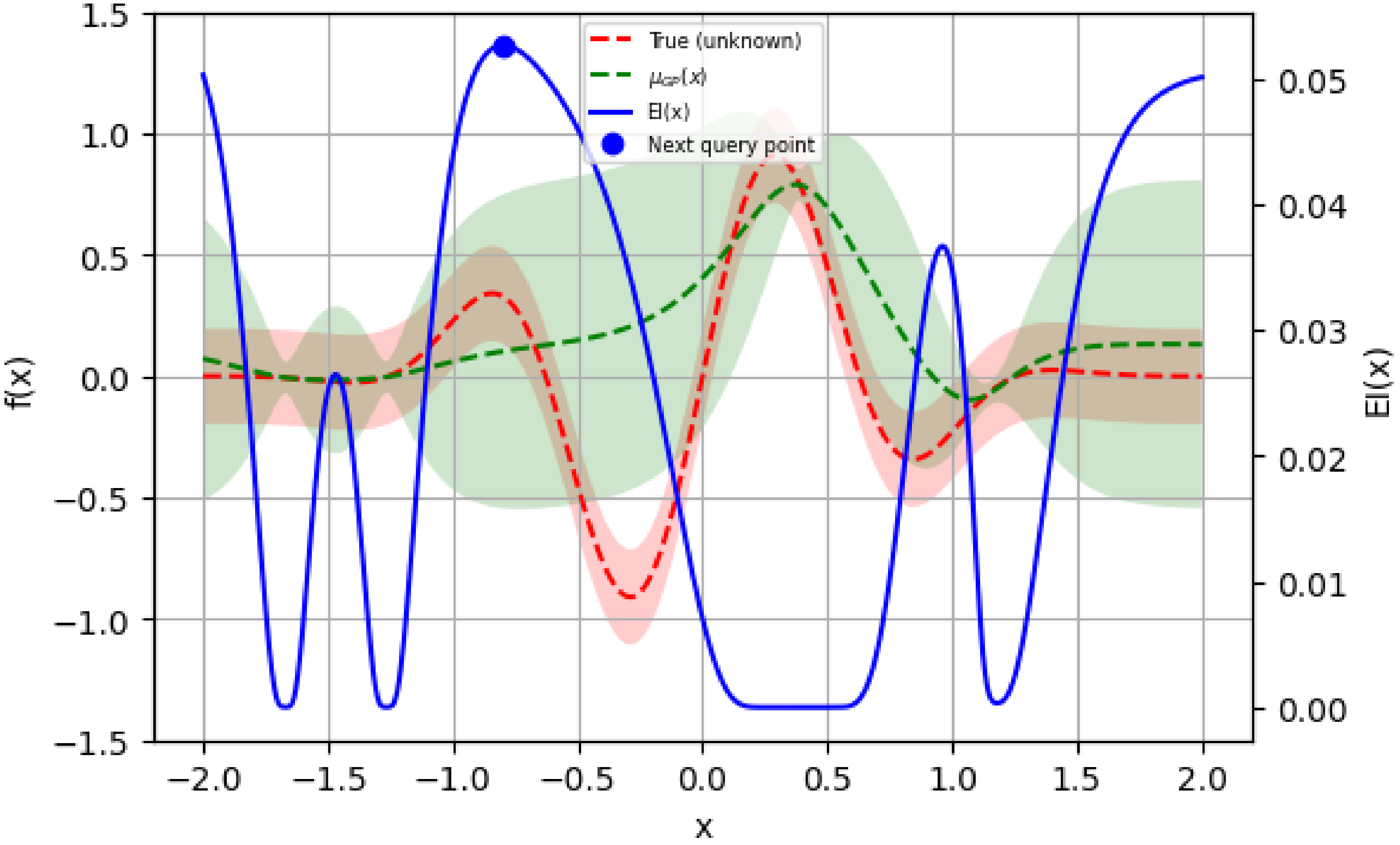
Bayesian Hyper Parameter Optimization

After Four Random Draws



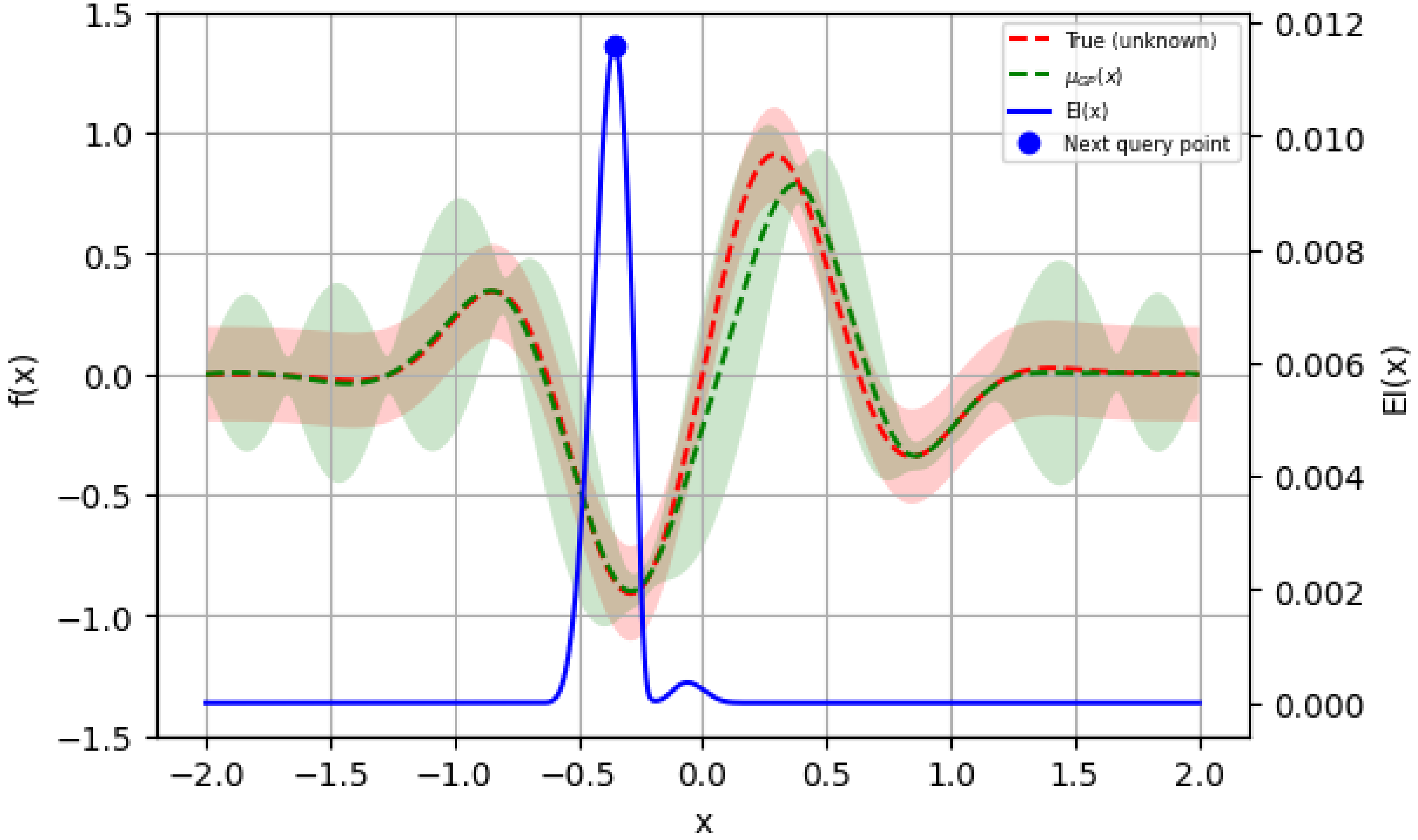
Bayesian Hyper Parameter Optimization

After One Smart Draw



Bayesian Hyper Parameter Optimization

After Ten Smart Draws



Bayesian Hyper Parameter Optimization

Key Ideas

- Learn a model to predict model score from hyper parameter values
  - Along with uncertainty around predicted score
  - Obtained scores at new hyper-parameter values are incorporated using Bayes theorem
- Evaluate the next hyper-parameter value that *could improve* the score the most
  - Results in explore-exploit balancing
  - Video explaining Bayesian Optimization
- Tools
  - `scikit-optimize`: integrates well with scikit-learn, very easy to use, mature software
  - `Hyperopt`: Newer, flexible, supports various algorithms suitable for complex search spaces.
  - `Optuna`: Newest, expands on techniques in Hyperopt but user-friendly, supports early-stopping while evaluating less promising hyper-parameters

  - Let’s see scikit-optimize in action (Lab7)